

Effective Lossless Compression of Sensor Information in Manufacturing Industry

Hagen Sparka, Roman Naumann, Stefan Dietzel, and Björn Scheuermann

Humboldt-Universität zu Berlin, Germany

Email: {sparkaha, roman.naumann, stefan.dietzel}@hu-berlin.de, scheuermann@informatik.hu-berlin.de

Abstract—In recent years, many visions for hitherto considered-futuristic computing applications gained momentum. The vision of smart factories is one example for these trends, which all share a common requirement: the prolific dissemination of sensor information. As wireless communication in smart factories needs to cope with harsh environments, the amount of sensor information produced by sources will likely surpass the communication channel's available capacity. This discrepancy calls for efficient communication and filtering protocols, as well as compression mechanisms, as a foundation for dependable applications. We propose such a compression algorithm that is lossless and tailored towards the requirements of the manufacturing industry. Our algorithm employs a two-step stochastic model that uses lossy compression to extract an approximation from the signal and a separate noise model to accommodate the remaining error. Evaluation results validate that our algorithm achieves better compression rates than existing approaches for several types of real world sensor data from the industry.

I. INTRODUCTION

Traditionally, production machines in the manufacturing industry were largely autonomous systems. Detailed information from the production process, when available at a machine, was used only locally by workers to monitor product quality and adherence to specifications. In the plastics industry, for example, sensorized molds are commonly used in production of safety-relevant parts. One core component of smart factory visions is to share and make available this process information to centralized systems for further exploitation. Possible applications include remote production process monitoring, creation of statistical models to predict future failures [1], and integration with business logic.

A key component in this endeavor is the timely transmission of information from several machines scattered throughout the factory to a central system. To this end, wireless transmission is a promising candidate, as it does not require factory retrofitting, i. e., laying additional network cables in the facility. Wireless communication in a factory is challenging, however, as it suffers from interference due to metal obstruction, insufficiently shielded machines, and moving obstacles such as forklift trucks [2], [3]. A number of network protocols have recently been proposed to improve robustness of wireless transmissions for smart factory use cases [4], [5]. While efficient and robust transmission is an important foundation, it does not solve the fundamental issue that the available sensor information generally surpasses the wireless channel capacity.

When process information is compressed before wireless transmission, bottlenecks during communication can be

avoided and more machines can be equipped with data-gathering hardware and operated simultaneously. General purpose lossless compression mechanisms, however, are suboptimal for the use case at hand, as they cannot exploit domain specific knowledge. In particular, sensor information often exhibits measurement noise, which is challenging for general purpose compression mechanisms. Using lossy compression to omit sensor noise is often not an option, because an exact representation of the acquired information, including noise, may be required for archival purposes and to solve product liability disputes. In general, it cannot be foreseen now what level of exactness future use cases require.

In this paper, we propose Two Step Model AC-Compressor (ThEMAtiC), a lossless compression mechanism that is tailored to sensor data acquisition use cases, especially in smart manufacturing settings. We hypothesize that sensor information can be effectively compressed by separating sensor noise from the sensor information curve's characteristic shape. As sensors typically observe certain physical quantities, their characteristic curve can be modeled much more efficiently when employing domain knowledge. By using domain-specific knowledge about typical production process information, we, therefore, achieve better compression than established general purpose approaches. More specifically, our proposed mechanism is entropy-optimal in the sense that we make optimal use of binary code word boundaries by employing Arithmetic Coding (AC). ThEMAtiC uses lossy compression based on the Discrete Cosine Transform (DCT) to separate (1) sensor noise and quantization error from (2) the characteristic shape of the sensor information curve that pertains to one production cycle. These two components of the signal form the basis of a stochastic model, which is used as an arithmetic coder's input to provide lossless compression. Separating the signal from noise and quantization error combined with applying domain knowledge to generate an efficient stochastic model for compression comprises our main contribution. This proposed algorithm achieves better compression than existing algorithms for several types of sensors commonly used in plastic injection molding, including particularly prevalent temperature sensors, pressure sensors, and injection screw position sensors. Our use case, plastic injection molding, is of special practical relevance, because it is one of the most widely used techniques in the plastics industry [6].

The remainder of this paper is organized as follows: first, we review related work in Section II. Next, Section III characterizes our primary use case and its associated process

information. We continue by providing a high-level overview of our proposed compression technique in Section IV and describe relevant algorithmic details in Section V. Section VI describes and discusses evaluation results using real-world sensor records. Section VII concludes the paper.

II. RELATED WORK

Recent results show that using artificial intelligence and genetic optimization algorithms can derive optimized process parameters to improve product quality and minimize mold setup time [7]–[9]. Huang [1], in particular, demonstrates that injection molding process parameters can be optimized effectively using recordings from sensors inside the cavity. These results, hence, motivate the necessity to efficiently transmit sensor information to be used for parameter optimization, amongst other use cases.

Artificial intelligence and genetic optimization algorithms can be applied to data compression, too. Here, they are used to optimize the parameters of a compression algorithm, for example, in lossy image compression [10], [11] or to choose a combination of different algorithms dynamically for each sample [12]. Further, genetic algorithms and neural networks can be incorporated in existing compression algorithms directly, for example, by replacing estimation modules with evolutionary optimization strategies [13] or a predictor based on a neural network like [14]. Though we do not use artificial intelligence or genetic optimization algorithms in this work, these techniques are highly promising candidates to further improve the compression algorithm presented in this work.

At first sight, the information dissemination requirements in our use case resemble those of wireless sensor networks (WSNs). WSNs are ad-hoc networks that consist of a high number of inexpensive, low-power nodes with sensors used for a wide number of applications, including health monitoring, disaster prediction, and military [15]. In the WSN context, a large body of literature exists on techniques that save bandwidth by means of summarization and inexpensive compression before transmission [16].

Marcelloni *et al.* [17], [18] propose an energy-efficient lossless compression algorithm for WSN applications that utilizes the high correlation of consecutive samples. An entropy encoder compresses these deltas using small, fixed Huffman [19] probability tables. Their approach is computationally inexpensive, as a fixed probability table is used and only two samples at a time need to be considered for compressing deltas. Kolo *et al.* [20] improve upon this idea by running two different entropy encoders in parallel. The currently best encoder is selected dynamically to improve compression ratio. Similar to our approach, Huang *et al.* [21] employ a DCT based noise filter to improve lossless compression of vibration data. The authors, however, do not derive a domain specific noise model and refrain from using a computationally more expensive entropy optimal encoding scheme. In general, WSN algorithms lay a strong focus on energy efficiency [16]. In the aforementioned algorithms, minimizing energy consumption is achieved by using fixed probability tables and, during

compression, only a constant number of samples at any point in time. In our use case, power is readily available at a machine; wireless channel capacity is still limited, though. Moreover, smart factories measure a wide variety of different physical quantities, and the recordings of different machines' sensors are less correlated [22].

As a basic building block for our algorithm, we employ Arithmetic Coding (AC) [23], which belongs to the family of entropy encoders, which also encompasses Huffman codes and run-length encoding [24]. Entropy encoders all have in common that they compress data losslessly based on an internal model of symbol probabilities [24]. A bad model for entropy encoders negatively impacts compression quality, whereas, as Gallager *et al.* [25] remarks, a precise model yields effective compression close to the bounds defined by Shannon's source coding theorem [26].

An entropy encoder assigns code words to input symbols. Huffman codes, for instance, use a simple bijection based on symbols' expected frequencies. This quite simple approach can result in good compression when the probability for each symbol s is $p(s) = 2^{-i}$ for some $i \in \mathbb{N}$. Arithmetic Coding (AC) performs equally well for such symbol distributions, but surpasses Huffman encoding when symbol probabilities adhere to a different distribution. Unlike Huffman encoding, AC does not represent input symbols by individual code words, but instead represents the whole input data as *one* joint code word, which is interpreted as a real number in the interval $[0, 1)$. This interval is subdivided for each symbol based on its respective cumulative probability. The result is a sub-interval $C \subset [0, 1)$; any code word $c \in C$ represents the original symbol sequence. AC, as described above, would require floating point arithmetic with infinite precision. Moffat *et al.* [27] describes several strategies to overcome this for practical implementations.

AC models used in the literature range from simple symbol frequency counts, so called zero-order models, and Markov chains to models where the last k symbols encoded are considered for calculating symbol probabilities [28], [29]. We employ AC for its closeness to the Shannon bound and the flexibility to design the stochastic model largely independent from the encoding process.

III. SYSTEM MODEL

Throughout this paper, we use the plastics industry as an example use case to show how to derive a compression model from domain-specific knowledge. Our proposed compression mechanism, however, is not restricted to this specific use case. By using our derivation process as a blueprint, similarly efficient compression strategies can be constructed for all kinds of applications where noisy sensor information is to be efficiently compressed without loss of exact information.

To apply ThEMAtiC directly, (1) the sensor's measurement should be subject to noise that resembles white noise, (2) the measured quantity should only change slowly with regard to the sample rate, and (3) individual recordings are not continuous to allow for DCT processing. Measurements matching

these properties can be found for example in physics, e. g., surface roughness measurements [30], medicine, e. g., functional magnetic resonance imaging [31], and biology/chemistry, e. g., cross sectional fluorescence profiles of organelles [32]. Further, when the properties mentioned above are not present, domain-specific knowledge may be used to either exchange the noise model or to choose a different approach to noise separation than that described in this work, for example, the Modified Discrete Cosine Transform for continuous sensor recordings [33].

For our example construction, we consider process information obtained during injection molding processes. Injection molding is one of the most commonly used production processes and not specific to the plastics industry. Generally speaking, it involves pumping (or “injecting”) melted material at very high pressures into a form, which is called “mold” or “tool.” Afterwards, the material cools down and solidifies, the mold is opened, and the finished part is taken out of the tool.

Relevant process information involves the injection screw position, machine temperature, and pressure and the temperature and pressure within the mold itself. In interviews with industry partners, we determined typical value ranges for different sensor types: temperature ranges between 1 °C and 1000 °C, pressure is 0 bar to 1000 bar. Typical sample rates range from 1 Hz to 1000 Hz; typical cycle durations are in the order of 1 s to 60 s. A machine usually has multiple sensors of each type, and factories host many machines, so the high sample rates can easily over-saturate wireless network capacity, necessitating some form of compression. Lastly, throughout this paper, we assume that measured sensor samples are discrete points, which is normally the case, as sensor signals are first quantized by analog-to-digital converters.

As numerous proposals for robust wireless communication protocols exist, we assume that machines have an, albeit slow or intermittent, connection towards the centralized server. In the following, we focus on using domain-specific knowledge to encode and compress information as efficiently as possible for transmission over this channel.

IV. COMPRESSION OVERVIEW

Our compression algorithm is based on arithmetic coding, a form of lossless entropy encoding. Arithmetic codes are a generalization of Huffman codes that do not impose restrictions on symbol probabilities for achieving near-optimal compression. Arithmetic coding requires a data model, i. e. symbol probabilities, as a parameter. The more precise the model, the better the compression that AC achieves. Figure 1 summarizes our AC-based compression: sensor samples and a data model are used as input to an arithmetic coder. Our main contribution is the derivation process for a data model that enables highly effective compression for typical industrial sensor information streams.

We derive our data model in two steps, which are shown in Fig. 2: first, a DCT is applied to the original sensor information. Exploiting that DCT results in a number of coefficients that represent information of the original signal with

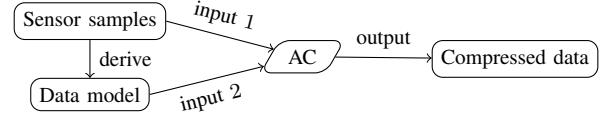


Fig. 1. Compression overview.

decreasing precision, we select a subset of the DCT’s output based on energy (not frequency) to obtain a lossy compression mechanism. Using the DCT in this way is similar but not identical to applying a low pass filter to the original signal: it approximates its basic shape, but filters noise components. The resulting overall compression algorithms, however, is lossless, as we compress the remaining error next.

Second, we model derivations – which represent sensor noise and approximation error – from this approximation using an additive white Gaussian noise (AWGN) model. The resulting data model consists of two components: (1) a subset of all DCT coefficients and their respective frequencies and (2) the AWGN variance. These two components together give us the probability that a sample at a given point in time assumes a certain value in the value range. Note that the way we choose the cosine coefficients here is especially well suited to separate the AWGN from the signal. In the frequency domain, the former will contribute uniformly to the energies of all frequencies, while the signal will produce a limited number of high energy coefficients.

Using AWGN to jointly model approximation error and sensor noise strikes an effective balance between complete customization of the probability distribution and using a generic distribution for all transmissions. The former would cause prohibitive overhead, as the whole table would need to be transmitted, whereas the latter would sacrifice compression performance by leaving domain knowledge unused.

Finally, we use the per-sample probability distributions and the unmodified sensor information as input to Arithmetic Coding. Both the resulting compressed sensor information and the data model are then sent over the wireless network to the receiver node. Using the received data model, this node reconstructs the per-sample probability distribution and applies arithmetic coding decompression to the compressed data.

V. MODEL DERIVATION

In this section, we discuss the model derivation process using our example use case in more detail. Our system model assumes that sensor readings are time-discrete samples from the domain $\mathbb{D} \subset \mathbb{R}$. We use the following vector notation to denote these samples.

$$\mathbf{x} = (x_1, x_2, \dots, x_n), \quad (1)$$

where for a particular sensor, n is the total number of obtained sensor values.

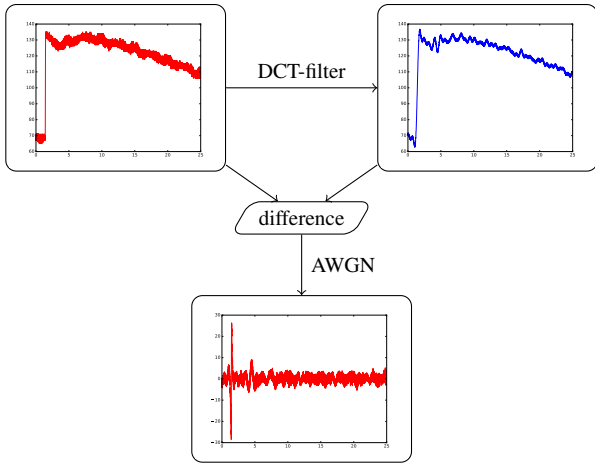


Fig. 2. Data model derivation.

A. Approximation separation

The first step of our model derivation algorithm is to derive an approximation of the original signal, which, effectively, filters out sensor noise while retaining the overall sensor curve shape. To this end, the Discrete Cosine Transform (DCT) and its inverse (IDCT) are employed, which provide mappings between time discrete sample vectors \mathbf{x} and a semantically equivalent set of n cosine coefficients $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)$. We define DCT and IDCT analogue to [34]:

$$\text{dct}(\mathbf{x})_k: Y_k = \sum_{n=1}^N x_n \cos\left(\frac{\pi(k-1)}{N}\left(n - \frac{1}{2}\right)\right), \quad (2)$$

$$\text{idct}(\mathbf{Y})_k: x_k = \frac{1}{2}Y_1 + \sum_{n=1}^{N-1} Y_{n+1} \cos\left(\frac{\pi n}{N}\left(k - \frac{1}{2}\right)\right). \quad (3)$$

The DCT provides energy compaction, which means that most of the signal's information resides in the spectrum's low frequency components. As a result, low frequency coefficients usually have larger absolute values than coefficients representing higher frequencies. The absolute coefficient values, however, are not monotonically decreasing as the frequency increases. Rather, a more-or-less constant residual energy indicates the presence of Gaussian noise, whereas irregular values represent the signal's true value or other, non-Gaussian noise. Therefore, we use a threshold to select the k most energetic coefficients, that is, the k coefficients with the greatest absolute value. The remaining coefficients exhibit an almost even distribution of energy throughout the spectrum, which corresponds to the definition of AWGN. By separating the signal's AWGN noise in this way, the resulting AC model achieves high precision. Our selection mechanism improves over other strategies, which transmit only low-frequency coefficients [22], select coefficients based on relative energy [21], or quantize high-frequency components with decreasing precision [35].

Formally, we define a selection function f_k that maps coefficient vectors from \mathbb{D}^n to an index set of size k such

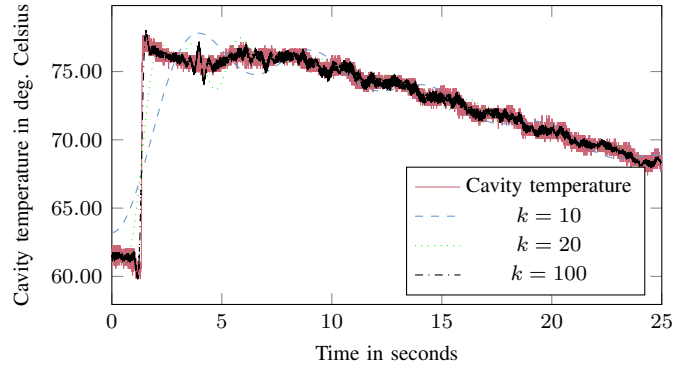


Fig. 3. Approximation $\hat{\mathbf{x}}$'s error for varying k .

that the sum of the selected coefficients' energy is maximal for a fixed k .

$$f_k(\mathbf{Y}) = \arg \max_{\substack{I \subseteq \{1, 2, \dots, n\} \\ |I|=k}} \sum_{i \in I} |Y_i|. \quad (4)$$

To generate the approximation without noise, we only use the coefficients selected by f_k and set remaining coefficients to zero:

$$\hat{Y}_i = \begin{cases} Y_i & \text{for } i \in f_k(\mathbf{Y}) \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

The resulting approximation $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$ is obtained by applying the IDCT to $\hat{\mathbf{Y}}$:

$$\hat{x}_k = \text{idct}(\hat{\mathbf{Y}})_k. \quad (6)$$

For larger values of k , the approximation $\hat{\mathbf{x}}$ more closely resembles the original sensor information \mathbf{x} , as can be seen in Fig. 3 for an example temperature sensor. Only $k \ll n$ coefficients, however, need to be transmitted to provide the approximation $\hat{\mathbf{x}}$.

B. Residue compression

Using only the DCT-based selection to reconstruct the original signal, our compression would be lossy. As ThEMAtIC is a lossless compression algorithm, we now introduce a method to compress the remaining error, that is, $\mathbf{x} - \hat{\mathbf{x}}$. More specifically, we introduce a model to derive near-optimal parameters for an Arithmetic Coding (AC) algorithm to compress the remaining error. We call this error the "residue" of the previous approximation.

We assume that, for a well-chosen system parameter k , the error $(\mathbf{x} - \hat{\mathbf{x}})$ is almost exclusively additive white Gaussian noise, since we removed energy peaks from the spectrum in the first step. Based on this assumption, we model the exact sample value at discrete time position i with a normal-distributed random variable

$$Z_i \sim \mathcal{N}(\mu_i, \sigma^2), \quad (7)$$

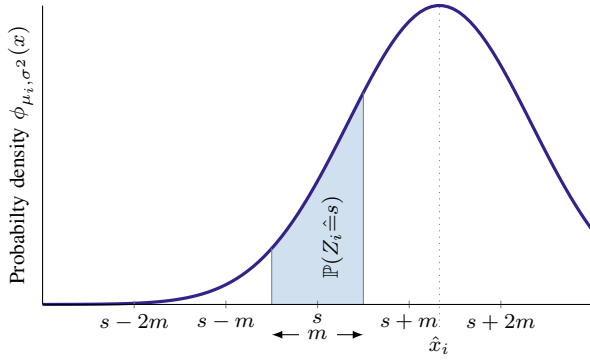


Fig. 4. Quantization-adjusted symbol probability for $\mu_i = \hat{x}_i$.

where $\mu_i = E[Z_i] = \hat{x}_i$ and σ^2 equals the sample variance,

$$\sigma^2 = \sum_{i=1}^n \frac{(x_i - \mu_i)^2}{n-1}. \quad (8)$$

For all practical purposes, sensor samples stem from a finite set of discrete values that is the result of quantization when analog sensor readings are digitized. For example, 65 536 unique values are possible if data points are represented using 16 bit precision. Thus, a specific measured sample does not correspond to a single real value, but instead represents a range of real data values. This quantization effect has to be considered when calculating the probabilities of the residues for AC compression. Here, we assume equidistant discrete sample points, where distances between measured values are multiples of m . When calculating the probability that x_i assumes value s , where $m|s$, we find the probability that Z_i is in the range $[s - \frac{m}{2}, s + \frac{m}{2})$. This probability is shown in Fig. 4 as the highlighted area under the normal distribution $\mathcal{N}(\mu_i, \sigma^2)$'s probability density function $\phi_{\mu_i, \sigma^2}(x)$, where the x-axis shows possible sample values. The marked probability can be calculated using the cumulative distribution function $\Phi_{\mu_i, \sigma^2}(x)$. Let

$$\mathbb{P}(Z_i \hat{=} s) = \Phi_{\mu_i, \sigma^2}(s + \frac{m}{2}) - \Phi_{\mu_i, \sigma^2}(s - \frac{m}{2}) \quad (9)$$

denote the symbol probability used by the arithmetic coder at each step to determine entropy-optimal binary code words for each $s \in \mathbb{D}$ given the corresponding approximated sample \hat{x}_i ($0 < i \leq n$) and the model's variance σ^2 . Note that using \hat{x}_i instead of the element in \mathbb{D} closest to \hat{x}_i as the normal distribution's center results in a more realistic model. This way, instead of applying AWGN to a quantized value, we model quantization of a noisy measurement.

C. Compression format

In Sections V-A and V-B, we saw how ThEMAtiC's stochastic data model is built in two steps: first, a subset of DCT coefficients is used to select characteristic features of the signal and second, the remaining approximation error is modeled with AWGN. This data model is used together with the original sensor information as input for an arithmetic coder.

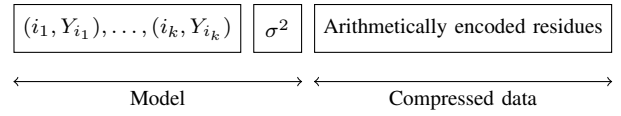


Fig. 5. Compressed data format.

The receiver needs to reconstruct the data model in order to reverse the compression. To this end, it requires the coefficients, their respective frequencies in the spectrum, and the AWGN model's variance. In addition, the receiver also requires the AC-encoded sensor samples. Putting all these components together, we obtain the data format that is shown schematically in Fig. 5: the leading tuples contain the coefficients $\{Y_i: i \in f_k(\mathbf{Y})\}$ and their frequency indices $i \in f_k(\mathbf{Y})$; together with the variance σ^2 , they constitute the stochastic data model. The AC-compressed residues follow afterwards, since their decompression requires prior reception of the model. Depending on the network's maximum transmission unit, the model and compressed data may need to be split into multiple packets before transmission.

D. Incremental decoding

An added benefit of transmitting the stochastic model parameters first is that the signal approximation is available before the whole compressed data is received. This allows for even more rapid decision making based on the sensor information's characteristic information. Analogously to the technique described in Section V-A, it is even feasible to decode an approximation before the whole model is received: when $J \subset f_k(\mathbf{Y})$ and coefficients $\{Y_i: i \in J\}$ have been received, the coefficient vector $\dot{\mathbf{Y}}$, where

$$\dot{\mathbf{Y}} = \begin{cases} Y_i & \text{for } i \in J \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

can be used to obtain the preliminary, i.e., partial, approximation $\dot{\mathbf{x}} = \text{idct}(\dot{\mathbf{Y}})$, which is less precise than $\hat{\mathbf{x}}$.

Obviously, incremental decoding is only beneficial when the model is sent in more than one packet. Even if network restrictions do not require splitting the model into multiple packets, smaller packets with a split model may help to improve otherwise low packet delivery rates in harsh environments [36].

E. Model compression

As can be seen in Fig. 5, the size of the stochastic model is dominated by the coefficient vector's size. Since the DCT gives us coefficients $Y_i \in \mathbb{R}$, conventional number representations, such as IEEE 754 floating point, are prohibitively large. To improve the overall compression ratio of our approach, it is therefore important to reduce the coefficients' binary size as much as possible. We found that the following combination of simple coefficient rounding and re-ordering constitutes an effective method of compressing the coefficient vector.

Since the DCT-provided filtering is merely an approximation, we can trade its precision with its binary size without necessarily negatively impacting overall compression ratio. We

found that representing coefficients with the same precision as sensor samples yields a good compromise between approximation error and model size. That is, we first round each coefficient Y_i for all $i \in f_k(\mathbf{Y})$ to its nearest discrete value in sample domain \mathbb{D} . In the second step, we reduce the number of bits needed for storing rounded coefficients by sorting coefficients in monotonically decreasing order by their absolute values: let l_1, l_2, \dots, l_k denote the sorted coefficients' indices so that $|Y_{l_u}| \geq |Y_{l_v}|$ holds $\forall u < v$ where $l_u, l_v \in f_k(\mathbf{Y})$; we then utilize this monotone ordering by allocating $\lceil \log_2 |\mathbb{D}| \rceil$ bits to Y_{l_1} , but only as many bits as were actually used by Y_{l_u} to coefficient $Y_{l_{u+1}}$ for all $1 \leq u < k$.

VI. EVALUATION

A. Methodology and datasets

To evaluate compression impact, we use real world sensor readings that were obtained during injection molding production processes. We compare ThEMAtiC's compression ratio with the WSN-specific compression algorithm SC [17], which is based on compressing information deltas, making it well-suited for compressing sensor information. In addition, we compare our algorithm against three computationally more expensive general-purpose compression algorithms: the Burrows-Wheeler-transformation-based [37] `bzip2`, the newer `xz` compression, which, among other techniques, also uses arithmetic coding, and the recently standardized [38] `brrotli` algorithm. For evaluation, we used the highest, i.e., most effective, compression level that each algorithm supports.

Unless otherwise noted, data points in plots of this section show the arithmetic mean, and error bars indicate 95% confidence intervals. Error bars might not always be visible in the figures when the error is very small.

The real-world dataset used in this evaluation is very heterogenous and contains a wide variety of subsets. It was recorded during injection molding of an automotive part, where the holding pressure was modified in a controlled way. Therefore the dataset not only contains recordings from "good" parts, but instead also includes recordings of parts with both aesthetic and technical issues. By varying the holding pressure, various defects such as short shots, sink marks, scratch marks, and overpacking were induced. Overall, our dataset is a collection of sensor recordings for more than 160 produced plastic parts, each containing the readings of five different sensors, measuring four different physical quantities with different noise characteristics and curve shapes. Here, we do not discern between the over 20 unique subsets forming our real-world dataset. Instead, we deem the presence of faulty parts an important property of our evaluation data, as the effective compression of their data is at least as important as compression of good parts.

During each cycle, five sensors record with 500 Hz sample rate each. The part produced is rather large with a 25 s cycle duration; thus, a total of 12 500 samples per sensor per cycle was collected. Two of the sensors were located at the injection molding machine and track the position of the screw that injects (1) plastic into the mold and (2) pressure in the

TABLE I
NUMBER OF SENSOR COEFFICIENTS

Sensor	k	k/n
Screw position	260	2.08%
Machine pressure	560	4.48%
Cavity pressure	620	4.96%
Cavity temperature front	440	3.52%
Cavity temperature back	380	3.04%

machine. The remaining three sensors are located in the mold's cavity and track (3) cavity pressure and its temperatures in the (4) front and (5) back. Temperature at more than one position can, for instance, be used to detect non-fills, that is, faults that stem from insufficient fill in the mold.

B. Compression model parameters

As described in Section V-A, ThEMAtiC uses a system parameter k that denotes the number of most energetic DCT coefficients used in the stochastic model's first step. Too few coefficients result in an imprecise approximation that impairs AC compression efficiency (cf. Fig. 3), whereas too many coefficients give a model that is too large to achieve good compression (cf. Section V-C). We chose k to optimize compression ratio on a per-sensor basis, but not per production cycle, so it is not required to transmit k as part of the model. In practice, deriving k can be done as a one-time set-up operation using, for example, historic sensor information.

The impact of transmitting k with the model would be low (approximately the same size as one coefficient index). We observed, however, that k differs little between production cycles and therefore chose to fix it per sensor and dataset; Table I shows the values for k that we use throughout our evaluation, both absolute and as the fraction of all coefficients.

C. Compression efficiency

Figure 6 compares compression effectiveness of ThEMAtiC to `brrotli`, `bzip2`, `xz`, and SC. The x -axis groups by sensor type, the y -axis shows compressed total size in bits divided by the number of samples n ; thus, a low number of bits per sample indicates a high compression ratio. In general, the energy-consumption-focused WSN algorithm SC is the least effective for all sensor types when compared to the computationally more complex general purpose algorithms, so we focus on comparing ThEMAtiC to those general purpose algorithms.

Our dataset's cavity temperature readings exhibited the highest degree of noise, so the good compression of ThEMAtiC matches our expectations: ThEMAtiC compresses cavity temperature between 16% and 29% better than the three general purpose algorithms.

For the cavity pressure sensor, the DCT requires a large number of coefficients to approximate the sensor readings with sufficient precision (see also Table I), resulting in performance comparable to but not surpassing that of generic compression mechanisms. For machine pressure, the achieved compression

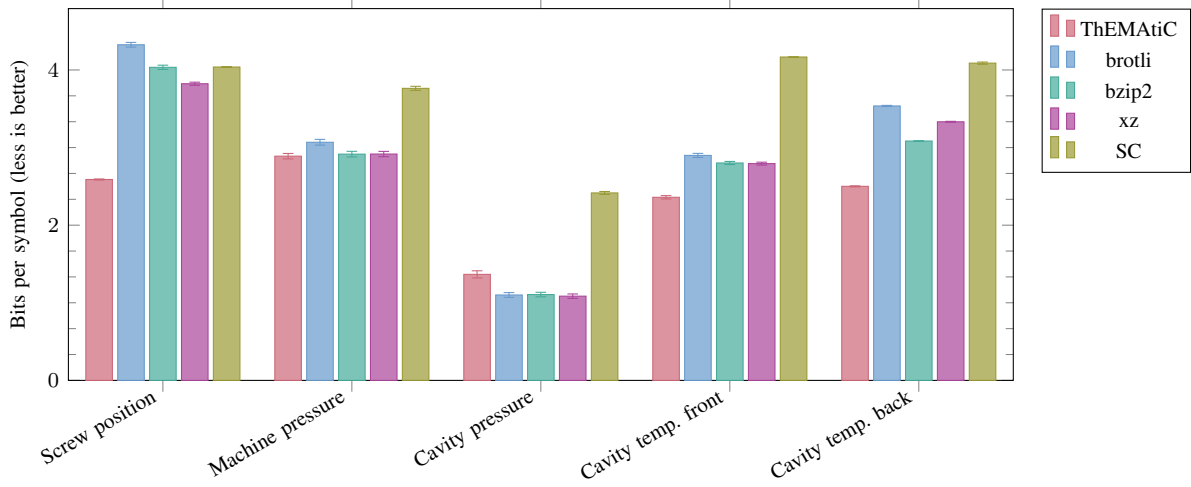


Fig. 6. Comparison of compression results.

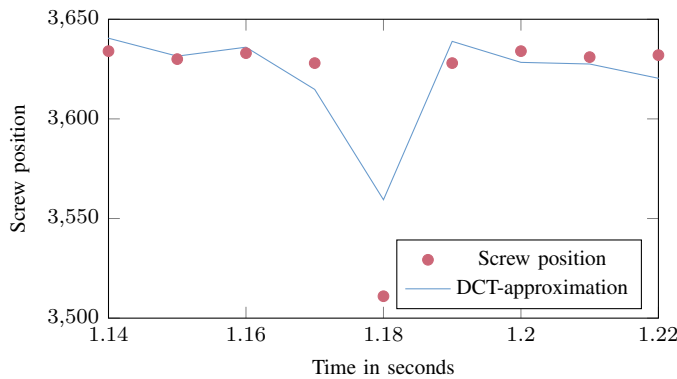


Fig. 7. Negative impact of periodic artifacts on the DCT-based approximation.

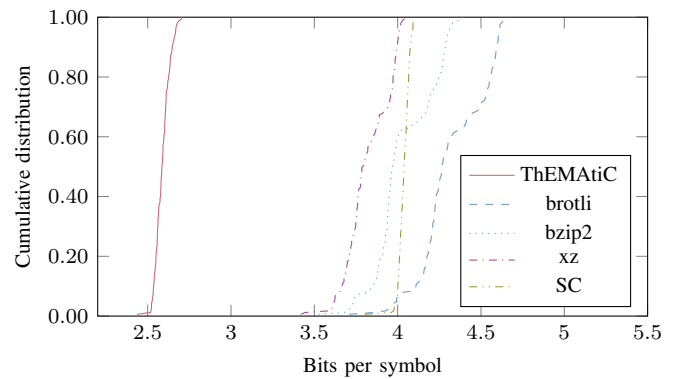


Fig. 8. Cumulative distribution of screw position sensor compression.

of ThEMAtiC is better than *brotli*. The difference in performance to the remaining two general purpose compression algorithms is statistically insignificant (for $p = 0.05$).

The machine-provided screw position sensor results in surprisingly efficient compression when using ThEMAtiC. Here, ThEMAtiC provides between 32% and 40% better compression than the other algorithms. The screw position sensor exhibits the lowest noise of all sensors, so we would expect that the separate residue approximation step in our stochastic model provides little benefit. The first step, however, requires only few coefficients on average to closely approximate the sensor signal, and the subsequent AWGN residue compression uses a low variance, which results in both a small model size and an effective AC compression.

Lastly, as our algorithm exploits domain-specific knowledge, we examine to what extent it provides benefits when sensor information does not fit our model assumptions. To this end, we use more than 100 production cycles of information obtained from a defective screw position *sensor*, not to be confused with correct sensor information for faulty injected

parts. Figure 7 shows an example of the defective sensor’s impact: the DCT approximation is unable to approximate the artifact at 1.18 s. Since the damaged sensor’s artifacts do not fit our Gaussian noise model, we would expect less effective compression, which proved only partially true: ThEMAtiC provides 4.5% to 22% better compression than the general purpose approaches, but it is 30% less effective than SC. We conclude that sensor-equipment quality and proper functioning are more important with our approach than when using general purpose algorithms. Albeit, the damaged sensor’s information is of limited practical use.

D. Compression distribution

We have seen that ThEMAtiC provides better average compression for several sensors. Now, we compare effectiveness for individual production cycles to assess whether compression performance is consistent or exhibits systematic biases for certain injection parameters. Figures 8 to 10 representatively show the cumulative distribution function for compression effectiveness in bits per symbol for the screw position sensor,

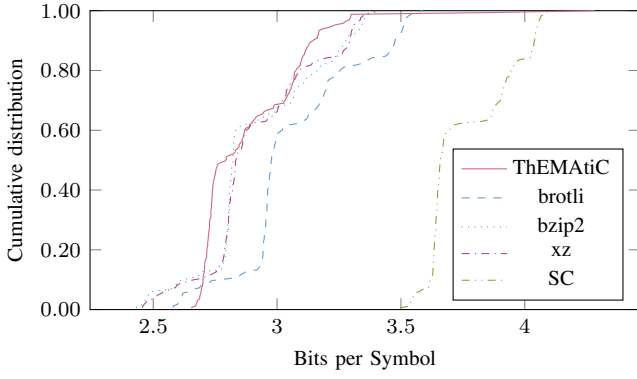


Fig. 9. Cumulative distribution of machine pressure sensor compression.

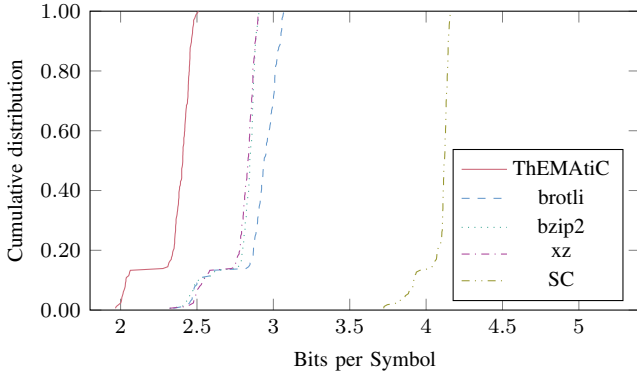


Fig. 10. Cumulative distribution of back cavity temp. sensor compression.

the back cavity temperature sensor, and the machine pressure sensor, respectively.

In both Figs. 8 and 10, it can be seen that ThEMAtiC consistently outperforms the other algorithms: for the screw position, each production cycles' sensor information's compressed form takes between 2.4 and 2.7 bits per symbol, which is 21% less than the best-compressed production cycle of each other algorithm.

Next, we consider the machine pressure sensor, where the different algorithms' results are very close to each other; the distribution is shown in Fig. 9. Here, ThEMAtiC's compression ratio is generally close to or better than the competing algorithms' and has the best median performance, but its compression is less effective than the general purpose approaches for two out of 165 production cycles, which is indicated by the "long-tail" in the upper region of Fig. 9. We found that these two production cycles show an unusually sharp peak in machine pressure that cannot be approximated well by the DCT. Despite the insufficient approximation of the DCT for these two production cycles, ThEMAtiC's compression effectiveness is still close to the general purpose algorithms' ranges and similar to SC.

Similar to the screw position results, the back cavity temperature is consistently compressed best using our ThEMAtiC compression. There is, however, more variation in the compression effectiveness: about 13% of the production

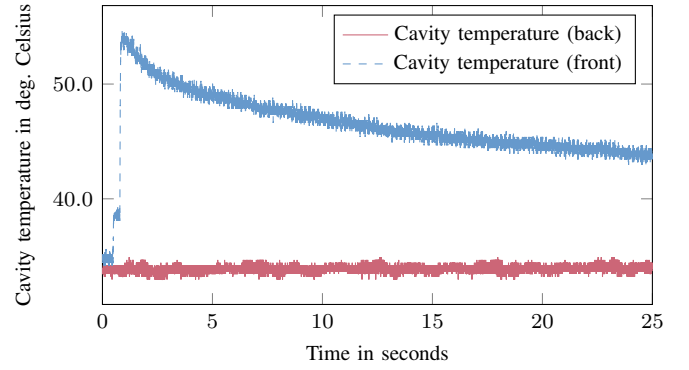


Fig. 11. A non-fill defect: cavity temperature.

cycles can be compressed with about two bits per symbol (in ThEMAtiC's case), whereas the remaining production cycles require 15% to 25% more space. Results of the generic compression mechanisms show a similar pattern. We found that these 13% of production cycles that can be compressed more efficiently are identical for all compression algorithms and correspond to a non-fill defect in the injected part. Figure 11 shows both front and back temperature sensors readings for an exemplary non-fill production cycle, where the material does not touch the back temperature sensor. Therefore, the back cavity temperature sensor shows nearly constant temperature and only noise has to be compressed, which is easier for all algorithms to a similar extent.

VII. CONCLUSION

Smart factories break with the traditional approach of operating largely autonomous systems in production machines. Wireless transmission is an enabler for this kind of machine-to-machine communication, but proves challenging due to limited channel capacity and harsh industrial environments.

We propose ThEMAtiC, a novel compression algorithm that exploits (1) the Discrete Cosine Transform's high degree of energy compaction on typical sensor information and (2) that typical sensor information is usually influenced by noise that resembles white noise. We make use of this knowledge by using a two-step stochastic modeling approach and by employing arithmetic coding to provide near-optimal entropy-encoded information. The result is a lossless compression algorithm that is tailored to industrial use cases.

The effectiveness of our compression algorithm is shown with real-world sensor recordings from the plastic industry. We considered more than 160 recordings of parts produced with five sensors active at the same time. Our results show that our domain-specific compression is up to 40 percent more effective than established general-purpose compression algorithms. Compression ratios were significantly improved for four out of five sensors. In a wireless transmission scenario, this means more machines can be equipped with sensor modules, or the same number of machines can sustain timely data transmission under more challenging network conditions.

ACKNOWLEDGMENTS

We wish to thank Francesc Bonada for kindly supplying and helping us with data from the injection-molding process. This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 636892. Website: www.PREVIEW-project.eu.

REFERENCES

- [1] M.-S. Huang, "Cavity pressure based grey prediction of the filling-to-packing switchover point for injection molding," *Journal of Materials Processing Technology*, 2–3 Mar. 23, 2007.
- [2] E. Tanghe, W. Joseph, L. Verloock, *et al.*, "The industrial indoor channel: Large-scale and temporal fading at 900, 2400, and 5200 MHz," *IEEE Trans. Wirel. Commun.*, Jul. 2008.
- [3] B. Milic, S. Brack, and R. Naumann, "Hierarchical configuration system for wireless mesh networks in manufacturing industry," in *Wireless and Mobile Networking Conference (WMNC), 2014 7th IFIP, IEEE*, 2014.
- [4] A. Willig, K. Matheus, and A. Wolisz, "Wireless Technology in Industrial Networks," *Proc. IEEE*, Jun. 2005.
- [5] D. Christin, P. S. Mogre, and M. Hollick, "Survey on Wireless Sensor Network Technologies for Industrial Automation: The Security and Quality of Service Perspectives," *Future Internet*, Apr. 8, 2010.
- [6] S. Kitayama, H. Miyakawa, M. Takano, *et al.*, "Multi-objective optimization of injection molding process parameters for short cycle time and warpage reduction using conformal cooling channel," *The International Journal of Advanced Manufacturing Technology*, Feb. 2017.
- [7] C. Shen, L. Wang, and Q. Li, "Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method," *Journal of Materials Processing Technology*, 2–3 Mar. 23, 2007.
- [8] A. Tellaeche and R. Arana, "Rapid Data Acquisition System for Complex Algorithm Testing in Plastic Molding Industry," in *Proceedings of World Academy of Science, Engineering and Technology*, World Academy of Science, Engineering and Technology (WASET), 2013.
- [9] W.-C. Chen, M.-H. Nguyen, W.-H. Chiu, *et al.*, "Optimization of the plastic injection molding process using the taguchi method, rsm, and hybrid ga-pso," *The International Journal of Advanced Manufacturing Technology*, Apr. 2016.
- [10] M. Horng, "Vector quantization using the firefly algorithm for image compression," *Expert Syst. Appl.*, 2012.
- [11] J.-T. Tsai, P.-Y. Chou, and J.-H. Chou, "Performance comparisons between pca-ea-lbg and pca-lbg-ea approaches in vq codebook generation for image compression," *International Journal of Electronics*, 2015.
- [12] S. Kargozar Nahavandy, P. Ghamisi, L. Kumar, *et al.*, "A Novel Adaptive Compression Technique for Dealing with Corrupt Bands and High Levels of Band Correlations in Hyperspectral Images Based on Binary Hybrid GA-PSO for Big Data Compression," *International Journal of Computer Applications*, Jan. 2015.
- [13] E. Cuevas, D. Zaldivar, M. A. P. Cisneros, *et al.*, "Block-matching algorithm based on differential evolution for motion estimation," *Eng. Appl. of AI*, 2013.
- [14] T. Mikolov, S. Kombrink, L. Burget, *et al.*, "Extensions of recurrent neural network language model," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011.
- [15] P. Rawat, K. D. Singh, H. Chaouchi, *et al.*, "Wireless sensor networks: A survey on recent developments and potential synergies," *The Journal of Supercomputing*, 2014.
- [16] M. A. Razzaque, C. J. Bleakley, and S. Dobson, "Compression in wireless sensor networks: A survey and comparative evaluation," *TOSN*, 2013.
- [17] F. Marcelloni and M. Vecchio, "A Simple Algorithm for Data Compression in Wireless Sensor Networks," *IEEE Commun. Lett.*, Jun. 2008.
- [18] F. Marcelloni and M. Vecchio, "An Efficient Lossless Compression Algorithm for Tiny Nodes of Monitoring Wireless Sensor Networks," *Comput J*, Nov. 1, 2009.
- [19] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proc. IRE*, 1952.
- [20] J. G. Kolo, S. A. Shanmugam, D. W. G. Lim, *et al.*, "An Adaptive Lossless Data Compression Scheme for Wireless Sensor Networks," *J. Sens.*, 2012.
- [21] Q. Huang, B. Tang, L. Deng, *et al.*, "A divide-and-compress lossless compression scheme for bearing vibration signals in wireless sensor networks," *Measurement*, 2015.
- [22] R. Naumann, S. Dietzel, and B. Scheuermann, "INFLATE: Incremental wireless transmission for sensor information in industrial environments," in *2015 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Dec. 2015.
- [23] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression," *Commun ACM*, Jun. 1987.
- [24] S. Saha, "Image Compression—from DCT to Wavelets: A Review," *Crossroads*, Mar. 2000.
- [25] R. Gallager and D. van Voorhis, "Optimal source codes for geometrically distributed integer alphabets (Corresp.)," *IEEE Trans. Inf. Theory*, Mar. 1975.
- [26] C. E. Shannon and W. Weaver, *A Mathematical Theory of Communication*. University of Illinois Press, 1963.
- [27] A. Moffat, R. M. Neal, and I. H. Witten, "Arithmetic Coding Revisited," *ACM Trans Inf Syst*, Jul. 1998.
- [28] A. Moffat, "Implementing the PPM data compression scheme," *IEEE Trans. Commun.*, Nov. 1990.
- [29] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Trans. Inf. Theory*, Jul. 1991.
- [30] C. Y. Poon and B. Bhushan, "Comparison of surface roughness measurements by stylus profiler, afm and non-contact optical profiler," *Wear*, 1995.
- [31] B. Biswal, E. A. Deyoe, and J. S. Hyde, "Reduction of physiological fluctuations in fmri using digital filters," *Magnetic Resonance in Medicine*, 1996.
- [32] M. Heilemann, S. van de Linde, M. Schüttelpelz, *et al.*, "Subdiffraction-resolution fluorescence imaging with conventional fluorescent probes," *Angewandte Chemie International Edition*, 2008.
- [33] J. Princen and A. Bradley, "Analysis/Synthesis filter bank design based on time domain aliasing cancellation," *IEEE Trans. Acoust. Speech Signal Process.*, Oct. 1986.
- [34] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *Comput. IEEE Trans. On*, 1974.
- [35] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, 1991.
- [36] J. Korhonen and Y. Wang, "Effect of packet size on loss rate and delay in wireless links," in *Wireless Communications and Networking Conference, 2005 IEEE, IEEE*, 2005.
- [37] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," 1994.
- [38] J. Alakuijala and Z. Szabadka, *Brotili Compressed Data Format*, RFC 7932 (Informational), Internet Engineering Task Force, Jul. 2016.